

## MODELOS DE DATOS

Un modelo de datos es un grupo de herramientas conceptuales para describir los datos, sus relaciones, su semántica y sus limitantes o restricciones de consistencia. El modelo de datos define la visión que tiene el sistema del mundo real y aporta la base conceptual para diseñar aplicaciones que hacen un uso intensivo de datos.

Un modelo de datos tiene dos componentes fundamentales:

- 1- El conjunto de reglas para representar las estructuras del mundo real o universo del discurso (parte estática).
- 2- El conjunto de operaciones autorizadas sobre las estructuras (parte dinámica).

Con respecto al diseño de bases de datos, el modelado de datos puede ser descrito así: "dados los requerimientos de información y proceso de una aplicación de uso intensivo de datos (por ejemplo, un sistema de información), construir una representación de la aplicación que capture las propiedades estáticas y dinámicas requeridas para dar soporte a los procesos deseados (por ejemplo, transacciones y consultas). Además de capturar las necesidades dadas en el momento de la etapa de diseño, la representación debe ser capaz de dar cabida a eventuales futuros requerimientos".

Un modelo de datos es por tanto una colección de conceptos bien definidos matemáticamente que ayudan a expresar las propiedades estáticas y dinámicas de una aplicación con un uso de datos intensivo. Conceptualmente, una aplicación puede ser caracterizada por:

- Propiedades estáticas: entidades (u objetos), propiedades (o atributos) de esas entidades, y relaciones entre esas entidades.
- Propiedades dinámicas: operaciones sobre entidades, sobre propiedades o relaciones entre operaciones.
- Las reglas de integridad sobre las entidades y las operaciones (por ejemplo, transacciones).

Así, un modelo de datos se distingue de otro por el tratamiento que da a estas categorías.

El resultado de un modelado de datos es una representación que tiene dos componentes: las propiedades estáticas se definen en un esquema y las propiedades dinámicas se definen como especificaciones de transacciones, consultas e informes.

Un esquema consiste en una definición de todos los tipos de objetos de la aplicación, incluyendo sus atributos, relaciones y restricciones estáticas. Correspondientemente, existirá un repositorio de información, la base de datos, que es una instancia o estado del esquema. Un determinado tipo de procesos sólo necesita acceder a un subconjunto predeterminado de entidades definidas en un esquema, por lo que este tipo de proceso puede requerir sólo un subconjunto de las propiedades estáticas del esquema general. A este subconjunto de propiedades estáticas se le denomina subesquema.

Una transacción consiste en diversas operaciones o acciones sobre las entidades del esquema o subesquema.

Una consulta se puede expresar como una expresión lógica sobre los objetos y relaciones definidos en el esquema; una consulta identifica un subconjunto de la base de datos.

Las herramientas que se usan para realizar las operaciones de definición de las propiedades estáticas y dinámicas de la base de datos son los Lenguajes de Definición de Datos (DDL), Lenguajes de Manipulación de Datos (DML) y los Lenguajes de Consulta (QL).

La investigación moderna sobre modelos de datos se ha centrado en los aspectos lógicos de las bases de datos y sobre los conceptos, herramientas y técnicas para el diseño de las mismas. Aspectos relativos a la implementación de los modelos, tales como velocidad de ejecución, concurrencia, integridad física y arquitecturas no son factores relevantes en el estadio de análisis de modelos de datos. La investigación más temprana sobre modelos de datos sí estaba más centrada en los aspectos de representación física.

### **Objetos permitidos por un modelo de datos**

Entidades: Empleado, cargo,...

Atributos: nombre, foto,...

Dominios: enteros reales,...

Interrelaciones: asociación entre empleado y cargo.

### **Operaciones permitidas por un modelo de datos**

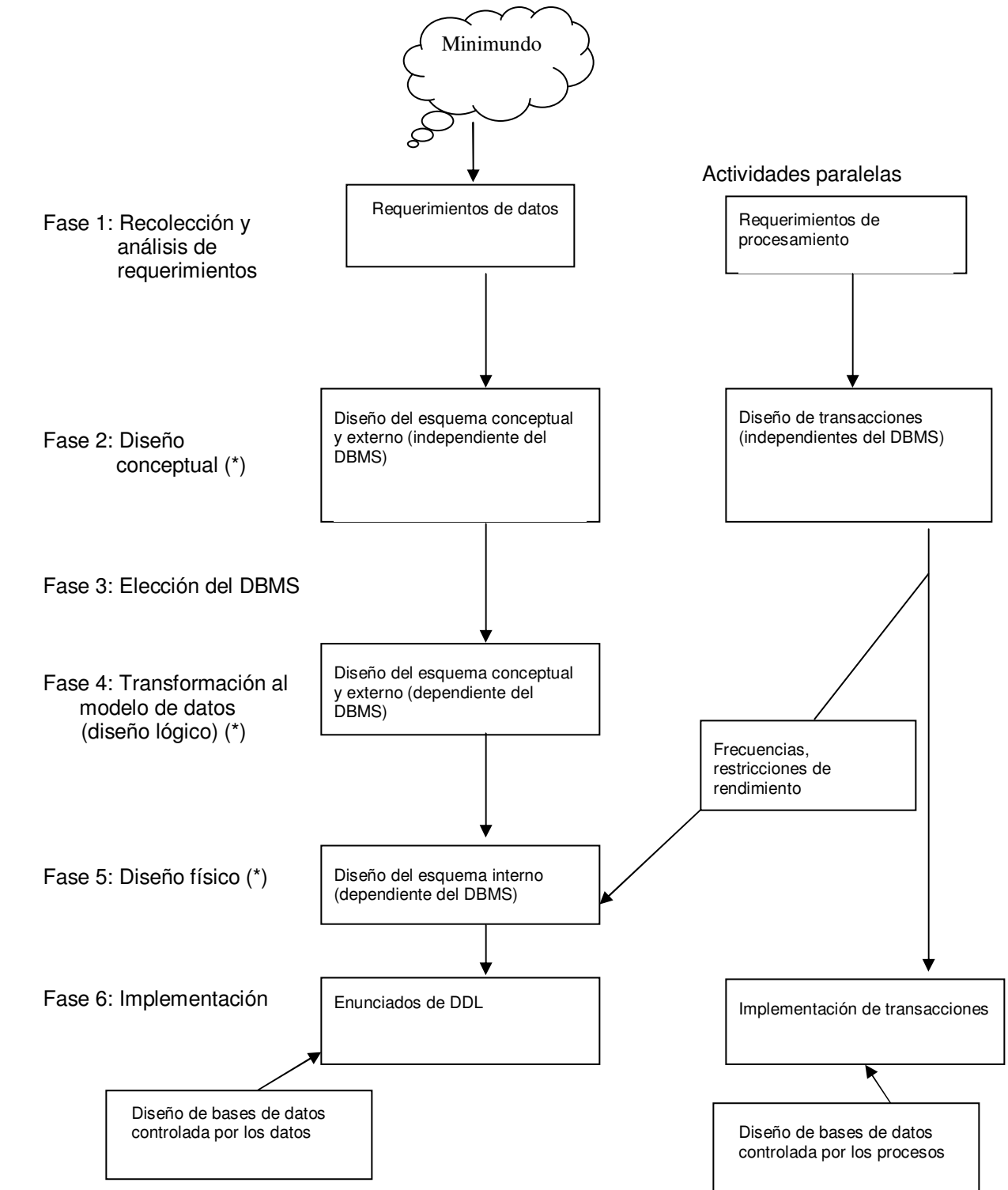
**Selección:** de entidades, atributos, interrelaciones, etc. que puede ser descrita con la ayuda de predicados que especifican las condiciones.

**Acción:** aplicada a entidades, atributos, interrelaciones, etc. que puede permitir recuperar o actualizar datos.

**Comportamiento:** Conjunto de operaciones definidas por el usuario válidas para la BD además de las operaciones básicas. Ej.: Para el objeto Estudiante el cálculo del promedio crédito.

El modelo de datos conceptual es el primer paso del proceso de desarrollo de bases de datos Top-Down y debe ser realizado durante la etapa de análisis del ciclo de desarrollo de un sistema de información.

## Fases en el diseño de bases de datos



(\*) Corazón del proceso de diseño

Se han propuesto varios modelos de datos:

- Los modelos conceptuales basados en objetos (de alto nivel).
- Los modelos lógicos basados en registros (de representación o de implementación).
- Los modelos físicos de datos (de bajo nivel): sirven para describir los datos en el nivel más bajo de abstracción (nivel interno). El diseño físico de la BD es el proceso de elegir estructuras de almacenamiento y caminos de acceso específicos para que los archivos de la BD tengan un buen rendimiento con las diversas aplicaciones de la BD.

## MODELOS CONCEPTUALES BASADOS EN OBJETOS

Se utilizan para describir los datos en los niveles conceptual y externo. Se caracterizan porque permiten una estructuración flexible y hacen posible una especificación clara de los limitantes de los datos.

Entre los modelos más representativos de este género están: el modelo Entidad-Relación (E-R) muy utilizado en la práctica y el modelo orientado a objetos.

### Descripción del modelo Entidad-Relación (E-R)

Es un modelo de datos semántico cuyo objetivo inicial era vencer algunas de las dificultades mostradas por el modelo relacional, al que pretendía sustituir. Concretamente, pretendía dotar de "significado" a las estructuras de datos, carentes del mismo, del modelo relacional.

En la práctica, este modelo de datos no ha llegado a implementarse en ningún DBMS comercial, pero ha tenido una enorme repercusión como herramienta de modelado de bases de datos (paradójicamente, bases de datos relacionales), existiendo hoy en día herramientas de diseño conceptual que incorporan la totalidad de sus conceptos e incluso productos que transforman diagramas conceptuales E/R en bases de datos reales en diversos formatos.

Consideramos que el modelo E-R se ha convertido en estándar para el diseño de bases de datos relacionales, por lo que lo utilizaremos para describir nuestra implementación.

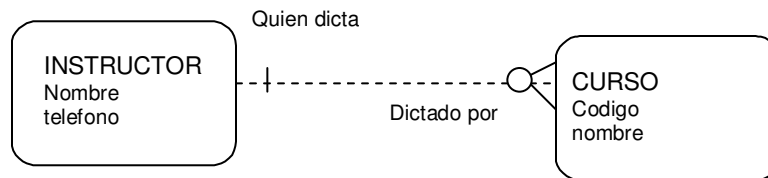
El modelo E-R percibe el mundo real/empresa/sección de la realidad, como un conjunto de objetos llamados *Entidades*, las cuales poseen *propiedades* (atributos) que las describen de alguna manera y un conjunto de asociaciones entre los objetos llamadas *Relaciones*.

Además este modelo representa ciertas limitantes que deben cumplir los datos contenidos en la BD; entre ellas la cardinalidad, que expresa el número de entidades con las que puede asociarse otra entidad mediante un conjunto de relaciones.

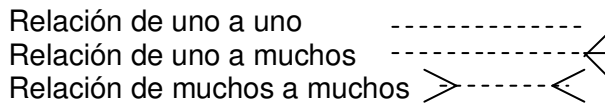
El modelo E-R aporta una herramienta de modelado para representar las entidades, propiedades y relaciones: los diagramas E-R. Mediante éstos, el esquema conceptual abstracto puede ser mostrado gráficamente y mantener una independencia conceptual

con respecto a la implementación propiamente dicha. En realidad, podemos hacer que los diagramas sean un reflejo fiel de las relaciones, interrelaciones y atributos del modelo relacional de datos o podemos englobar diversas relaciones en una sola entidad o conjunto de propiedades.

La figura muestra un ejemplo que nos servirá para mostrar cómo han de interpretarse estos diagramas.



Los **rectángulos** representan Entidades, dentro del rectángulo en la parte superior se coloca el nombre de la entidad y en la parte inferior se colocan las **propiedades** de la entidad. El tipo de relación entre dos entidades se representa mediante la siguiente simbología:



## Descripción del modelo orientado a objetos

Está basado en el paradigma de la programación orientada a objetos.

Se presenta una relación entre los términos OO y los términos de programación tradicional.

Término OO	Término en programación
Objeto	Variable
Clase	Tipo
Método	Función
Mensaje	Llamada
Jerarquía de Clase	Jerarquía de Tipos

Al igual que el modelo E-R, el modelo orientado a objetos se basa en una colección de objetos.

Un **objeto** es cualquier cosa real o abstracta acerca de la cual almacenamos datos y los métodos que controlan dichos datos. Ej.: Cuenta (nro, saldo, pago\_interes); Objeto\_geométrico (forma, área). Un objeto en la terminología OO corresponde mas o menos a una variable en el sentido que tiene en programación; por tanto, tiene un tipo (el termino en OO es Clase) y tiene un valor (el termino en OO es Estado).

Los objetos pueden ser tan sencillos o complejos como lo deseemos. En primer lugar, el sistema ofrecerá un conjunto de clases primitivas (Tipos en términos de programación tradicional) de objetos ya integrados, como por ejemplo INTEGER (entero) o FLOAT (número de punto flotante). El usuario puede crear objetos individuales de esas clases primitivas.

Si un objeto está formado por otros objetos, esos objetos componentes se denominan Variables de Caso, por ejemplo, un objeto “Tupla de Clientes” contiene tres variables de caso llamadas `Id_cli`, `Nom_cli`, `fech_nac_cli`.

Los objetos que contienen los mismos tipos de valores y los mismos métodos se agrupan en **clases**. El término OO “Clase” corresponde aproximadamente a la idea tradicional de Tipo de Datos, porque lo importante es que:

- Los usuarios puedan definir sus propias clases.
- Todas las clases definidas por el usuario o integradas, llevan consigo un conocimiento de los operadores aplicables a los objetos de esa clase (el término OO es “Método”).

La única forma de operar sobre un objeto es mediante los operadores (Métodos) definidos por la clase de ese objeto. Por ejemplo, podríamos tener una clase definida por el usuario llamado DEPARTAMENTOS, con los métodos ADICIONAR\_DEP, CAMBIAR\_DEP, así, estos métodos solo pueden aplicarse a objetos DEPARTAMENTO individuales.

La ventaja del sistema anterior es que hace posible modificar la representación interna de los objetos DEPARTAMENTO sin tener que volver a escribir ninguna de las aplicaciones que utilizan esos objetos, desde luego, siempre que estos cambios en la representación interna vayan acompañados de una modificación correspondiente en la definición de los métodos aplicables a los objetos DEPARTAMENTO.

Un **método** es en esencia una operación o función que puede realizarse con los objetos de alguna clase específica. Podemos considerar que el conjunto de métodos aplicables a una clase dada se almacena junto con la definición de esta clase; juntos, definen la “Interfaz Pública” de los objetos de esa clase. Los objetos pueden actualizarse o examinarse sólo a través de su interfaz pública.

Los **mensajes** son solicitudes para que se lleven a cabo operaciones indicadas y se produzcan resultados. Para aplicar un método dado a un objeto determinado es necesario enviar un mensaje a ese objeto. Al recibir el mensaje, el objeto ejecuta la función (o sea, el método) solicitada por el mensaje, y en seguida devuelve un resultado al remitente.

### Jerarquía de Clases

Si la clase B es una subclase de la clase A, todo caso de B será en forma automática un caso de A, pero lo opuesto no se cumple. Por ejemplo, podríamos tener una clase llamada EMPLEADOS, y otra clase llamada PROGRAMADOR la cual es una subclase de EMPLEADO (todo programador es un empleado, pero no viceversa).

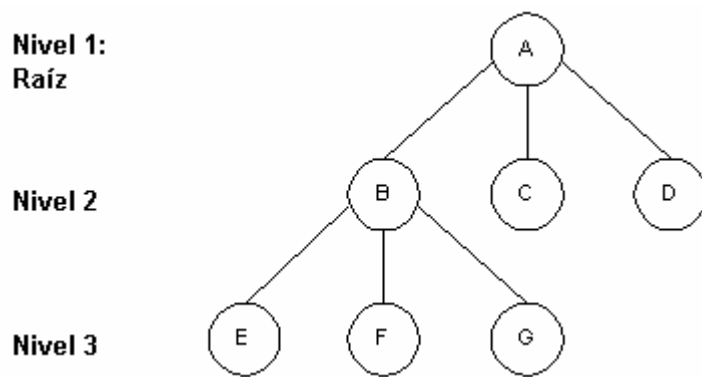
## MODELOS LÓGICOS BASADOS EN REGISTROS

Se utilizan para describir los datos en los niveles conceptual y externo. A diferencia de los modelos de datos basados en objetos, estos modelos sirven para especificar la estructura lógica general de la BD; sin embargo no permiten especificar en forma clara los limitantes de los datos.

De estos modelos los más ampliamente aceptados son: El modelo jerárquico, el modelo de red y el modelo relacional.

### Descripción del Modelo jerárquico

Un DBMS jerárquico utiliza jerarquías o árboles para la representación lógica de los datos. Los archivos son organizados en jerarquías, y normalmente cada uno de ellos se corresponde con una de las entidades de la base de datos. Los árboles jerárquicos se representan de forma invertida, con la raíz hacia arriba y las hojas hacia abajo.

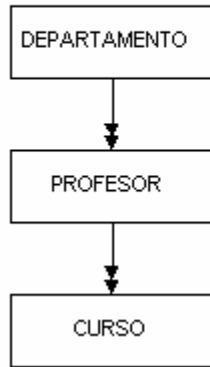


Un DBMS jerárquico recorre los distintos nodos de un árbol en un *preorden* que requiere tres pasos:

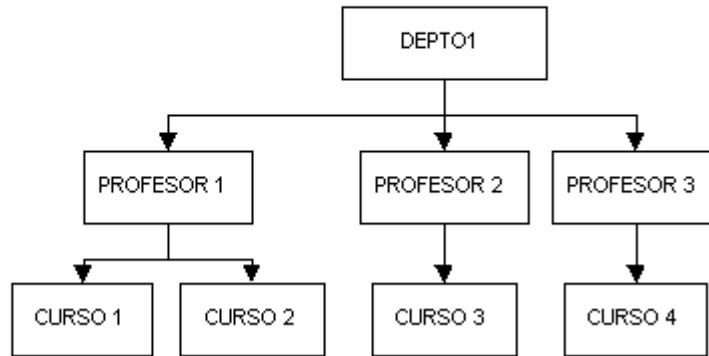
1. Visitar la raíz.
2. Visitar el hijo más a la izquierda, si lo hubiera, que no haya sido visitado.
3. Si todos los descendientes del segmento considerado se han visitado, volver a su padre e ir al punto 1.

Cada nodo del árbol representa un tipo de registro conceptual, es decir, una entidad. A su vez, cada registro o segmento está constituido por un número de campos que los describen – las propiedades o atributos de las entidades. Las relaciones entre entidades están representadas por las ramas (internamente apuntadores). En la figura cada departamento es una entidad que mantiene una relación de uno a muchos con los profesores, que a su vez mantienen una relación de uno a muchos con los cursos que imparten.

### Estructura lógica



### Ejemplo de base de datos



A modo de resumen, enumeramos las siguientes características de las bases de datos jerárquicas:

1. Los segmentos de un archivo jerárquico están dispuestos en forma de árbol.
2. Los segmentos están enlazados mediante relaciones uno a muchos.
3. Cada nodo consta de uno o más campos.
4. Cada ocurrencia de un registro padre puede tener distinto número de ocurrencias de registros hijos.
5. Cuando se elimina un registro padre se deben eliminar todos los registros hijos (integridad de los datos).
6. Todo registro hijo debe tener un único registro padre excepto la raíz.

### Descripción del Modelo de red

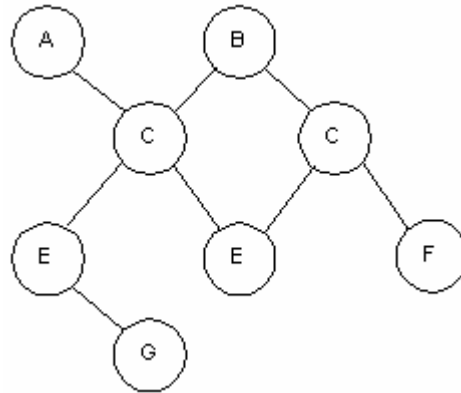
El modelo de red intenta superar las deficiencias del enfoque jerárquico, permitiendo el tipo de relaciones de muchos a muchos.

Una estructura de datos en red, o estructura *plex*, es muy similar a una estructura jerárquica, de hecho no es más que un superconjunto de ésta. Al igual que en la estructura jerárquica, cada nodo puede tener varios hijos pero, a diferencia de ésta, también puede tener varios padres. La figura muestra una disposición plex. En esta representación, los nodos C y E tienen dos padres, mientras que los nodos F, G tienen sólo uno.

El concepto básico en el enfoque de red es el *conjunto* ('set'), un conjunto está constituido por dos tipos de registros que mantienen una relación de muchos a muchos.

Para conseguir representar este tipo de relación es necesario que los dos tipos de registros estén interconectados por medio de un registro conector llamado *conjunto conector*.





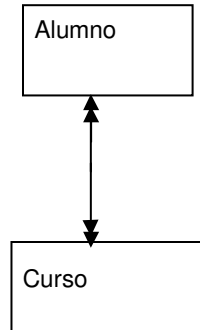
Los conjuntos poseen las siguientes características:

- El registro padre se denomina propietario del conjunto, mientras que el registro hijo se denomina miembro.
- Un conjunto está formado por un solo registro propietario y uno o más registros miembros.
- Una ocurrencia de conjuntos es una colección de registros, uno de ellos es el propietario y los otros los miembros.
- Todos los registros propietarios de ocurrencias del mismo tipo de conjunto deben ser del mismo tipo de registro.
- El tipo de registro propietario de un tipo de conjunto debe ser distinto de los tipos del registro miembro.
- Sólo se permite que un registro miembro aparezca una vez en las ocurrencias de conjuntos del mismo tipo.
- Un registro miembro puede asociarse con más de un propietario, es decir, puede pertenecer al mismo tiempo a dos o más tipos de conjuntos distintos. Esta situación se puede representar por medio de una estructura multianillo.
- Se pueden definir niveles múltiples de jerarquías donde un tipo de registro puede ser miembro en un conjunto y al mismo tiempo propietario en otro conjunto diferente.

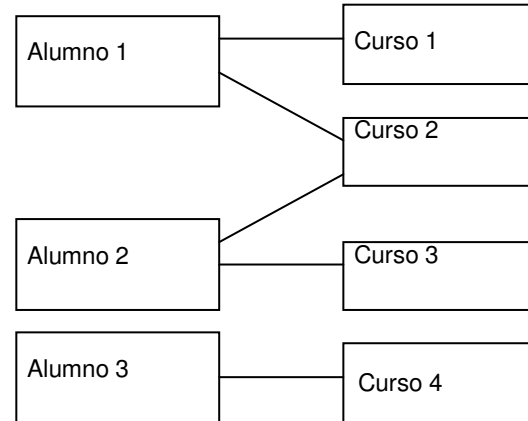
Los registros de la BD se organizan en forma de conjuntos de gráficas arbitrarias.

En la figura cada Alumno es una entidad que mantiene una relación de uno a muchos con las Curso, que a su vez mantiene una relación de uno a muchos con los Alumno.

### Estructura lógica



### Ejemplo de base de datos



## Descripción del Modelo relacional

El modelo relacional de datos supuso un gran avance con respecto a los modelos anteriores. Este modelo está basado en el concepto de relación. Una relación es un conjunto de n-tuplas. Una tupla, al contrario que un segmento, puede representar tanto entidades como interrelaciones N:M. Los lenguajes matemáticos sobre los que se asienta el modelo relacional, el álgebra y el cálculo relacionales, aportan un sistema de acceso y consultas orientado al conjunto. La repercusión del modelo en los DBMSs comerciales actuales ha sido enorme, estando hoy en día la gran mayoría de los gestores de bases de datos basados en mayor o menor medida en el modelo relacional.

El concepto de *modelo de datos* en sí, surgió al mismo tiempo que el modelo relacional de datos fuera propuesto por su creador Ted Codd, después de que los modelos jerárquico y de red estuvieran en uso. Posteriormente, estos dos modelos fueron definidos independientemente de los lenguajes y sistemas usados para implementarlos. Con anterioridad no eran más que colecciones de estructuras de datos y lenguajes sin una teoría subyacente definida.

En cuanto al modelo relacional, no se puede decir que sea en sí un modelo semántico de datos. Su enorme éxito no se debe a que permite de forma implícita operaciones conceptualmente abstractas sobre los datos, sino a los altos niveles de fiabilidad e integridad que aporta en el manejo de grandes cantidades de datos.

Desde su comienzo en 1970 y durante mucho tiempo después, los sistemas gestores de bases de datos relacionales (RDBMS : Relational Database Management System) estuvieron restringidos al ámbito de los mainframes y mini-computadores. Con la irrupción masiva en el mercado de los micro-computadores, aparecieron algunas implementaciones de RDBMSs que intentaban emular las propiedades de los grandes sistemas, aunque no contaban con la mayor parte de las características necesarias para

ser denominados "relacionales", especialmente en lo que se refiere al cumplimiento de las reglas de integridad relacional.

Hoy en día contamos con RDBMSs para micro-computadores que sí pueden ser considerados plenamente relacionales y que, si bien no llegan alcanzar las prestaciones de los grandes sistemas en cuanto a velocidad de ejecución, seguridad, integridad de datos, recuperación y estabilidad, no tienen nada que envidiar a éstos cualitativamente, y sus deficiencias se deben sobre todo al tipo de máquina en el que funcionan y a los sistemas operativos que estas máquinas utilizan.

Lo que realmente marca la diferencia entre los sistemas relacionales y los sistemas anteriores es el hecho de que su creador, Ted Codd, basó expresamente su funcionamiento sobre un modelo matemático muy específico: el álgebra relacional y el cálculo relacional, así como la progresiva adopción, por parte de su creador y algunos colaboradores, de un número de Reglas de Integridad Relacional y de Formas Normales.

En este modelo los datos y las relaciones entre los datos se representan por una serie de tablas, cada una de las cuales tiene varias columnas con nombres únicos. Este modelo conecta registros mediante valores de campos. Ej.: BD que incluye a los Profesores y sus Cursos.

PROFESOR

Cedula	Nombre	Ciudad
20	Juán	Medellín
30	Luis	Bogotá
40	Ana	Cali

CURSO

Codigo	Nombre	Cedula
02	A1	20
03	A2	30
04	A3	20
05	A4	40